CHAPTER 8

COMPLETING THE BASIC DESIGN

We accomplished a great deal in Chapter 7 – we took a product idea to a working prototype. But there is a little more design work to do such that our prototype can be called a “USB I/O device”. Additionally, if we want to move our prototype out of the lab and into the market place there are several more steps that we have to make. This chapter covers the requirements of building a USB **complaint** I/O device and the assistance that the USB Implementers Forum will provide to ensure that your product is sellable throughout the world.

The additional design steps involve the power handling of our I/O device while the marketing of our product involves a Vendor ID. The USB IF also supplies a suite of tests to enable thorough testing of your product.

HANDLING POWER REQUIREMENTS

Let’s recap the USB Specifications Power Requirements – there are two distinct operating modes of an I/O device, active and suspended. When active our I/O device can draw up to 100mA from Vbus and be classified as a low power device and it can draw up to 500mA from Vbus and be classified as a high power device. My recommendation is to strive to get under the 100mA limit since your device will be operational even on an unpowered hub. If your I/O device requires more than 500mA then it must be self-powered and will therefore have its own power cord.

There are also limits on the amount of current that an I/O device can draw when it is suspended. But, first of all, what does “suspended” really mean? A modern PC, and most laptop computers, will turn themselves into a sleep state if they are not used for some pre-defined period of time. This tactic saves power on a desktop machine and preserves battery life on a laptop computer. There are various levels of “sleep” but the one preferred by most manufacturers is a low-power “suspend to RAM” where the PC can quickly wakeup and be available for computing within a few seconds (the documentation calls this the S3 state).

So a modern PC will suspend itself when not being used. If the PC host is not active then there is no reason for most of the I/O devices to be active either – in fact, a USB I/O device is REQUIRED to enter a suspend state when instructed to do so by the PC host. A PC host will signal that it is about to enter a suspend state by stopping the transmission of SOF packets – when an I/O device detects that three consecutive SOF packets have been missed then it starts to suspend itself. Note that Vbus is still present but the current drain per device is reduced to 500uA maximum – this is a design challenge, as we shall see in the following example.

Some I/O devices will want to wakeup the PC host; a telephone, for example, will need to wakeup the PC host so that it can answer the telephone. We shall design this capability into our example I/O device so that you can understand the process and the techniques involved. The USB specification allows up to 2.5mA for an I/O device that is capable of generating a wakeup sequence, again, this is not a lot of power to implement the function.

RESPONDING TO A SUSPEND

All USB I/O controllers have some built-in mechanism to detect the lack of signaling on USB which signifies a suspend request. The EZ-USB microcontroller used in our "buttons and lights" example, for instance, generates an internal interrupt to the microcontroller. Within this interrupt service routine the microcontroller does whatever is necessary to reduce its power consumption from Vbus as shown in Figure 8.1. In our example it will ensure that all of the LEDs are OFF and will then enter a low-power state (it turns off clocks to most of its internal circuitry). Other microcontrollers implement similar mechanisms since this is required by the USB specification.

<<paste firmware here>>

Figure 8.1 Responding to a Suspend Request

Generating a Wakeup Event

Any I/O device that wishes to generate a wakeup event must pre-declare this ability in its Configuration Descriptor (see Chapter 3). This declaration is required so that the operating system can create a software process to monitor and control this operation. The I/O device is also required to have a feature report interface that allows the PC host to enable and disable this ability.

While suspended, an idle USB is in the “J” state. An I/O device signals a wakeup event by driving the bus to the “K” state for 10-15 msec. Note that this is the ONLY situation where signaling is initiated from the I/O device, all other transactions are initiated by the PC host. Any hubs between the I/O device and the PC host will recognize this constant K State and will propagate it towards the PC host. The PC Host controller will respond to this Wakeup Event.

Not all USB microcontrollers have the ability to signal a wakeup event. The EZ-USB microcontroller does – an external pin first brings the EZ-USB out of its deep sleep state and then signals a wakeup (the EZ-USB documentation calls this a RESUME). The interrupt service routine is shown in Figure 8.2.

<<paste firmware here>>

Figure 8.2 Generating a wakeup event.

With the addition of the Suspend and Wakeup interrupt service routines our buttons and lights example now complies with the USB specifications Power Dissipation Requirements. A small addition to the initialization routine to enable these interrupt levels was also required in our EZ-USB example.

So we are now ready to move our I/O device out of our lab and into the world. There is a problem – the example uses a Vendor ID of 4242H which is registered to Intel as "USB Design by Example". **You** are not allowed to sell a product with this vendor ID (I could, since I own it).

If you intend to sell your I/O device then you must obtain your own Vendor ID.

GETTING A VENDOR ID

There are two methods that you can use to get your own Vendor ID and both are administered by the USB Implementers Forum (USB-IF), a non-profit organization chartered to promote the Universal Serial Bus. The first method is a simple one-time purchase of a number and this costs $200.00. The second is to join the USB IF as a member and one of the many benefits of being a member includes a Vendor ID. This option costs $2500/annum. Both methods start with an email to admin@usb.org – the equivalent of a hot-line number.

The USB specification, its revisions and a variety of complementary documentation such as CLASS Specifications and TEST PROCEDURES are managed by the USB-IF. The USB IF run a web-site at www.usb.org and also organizes many member events. There are regular Developers Forums where members hear the latest technical disclosures on advancements in USB including software driver support. USB-IF member booths are organized at venues such as Comdex and Intel's Developer Forum where exposure to customers and new products are made. The USB-IF also organizes quarterly "Plugfests".

A typical Plugfest lasts three days and member companies bring USB products (host, hub, I/O device, software stacks) to be tested. The tests measure compliance to the USB specification and interoperability with other USB products. The remainder of this chapter will present the tests and fixes for common problems. Much of the testing software that will be described is available at the www.usb.org web site and diligent pre-testing before attendance at a Plugfest will produce a more successful testing experience. Products that pass all tests are added to a USB IF Integrators List that is available to member companies.

The Universal Serial Bus has become a dominant PC peripheral interconnect bus. This is due, at least in part, to industry efforts to improve the overall quality and interoperability of the USB devices on the market. There are many factors, from borderline electrical signaling to poorly written device drivers, that may cause a device to work just fine with one host controller and set of peripherals, but fail to work or cause other devices to fail to work in a different system or scenario. It can be a daunting task, but trying to address these factors, as best as possible, is critical for the success of a mainstream PC bus. Virtually all devices need to interact and function together for a mainstream bus to be successful.

The remainder of this chapter discusses the major industry USB quality certification programs that are available for I/O device vendors. In particular, the USB compliance testing performed by the USB Implementors Forum (USB-IF) and the Microsoft Windows Hardware Quality Labs (WHQL) are presented. I must thank Dan Froelich of Intel's Architecture Labs for his major contributions to this chapter.

This chapter covers the following topics:

An overview of the USB-IF compliance program and compliance logos.

A technical summary of USB-IF compliance tests.

An overview of Microsoft WHQL USB certification.

A technical summary of Microsoft WHQL USB testing.

Finally I present some of the common issues that are uncovered by the testing.

USB-IF Compliance Overview

The USB-IF have conducted USB Compliance Testing Workshops 4-5 times a year since 1996. These workshops focus on testing basic device electrical characteristics, device firmware, and the general interoperability between the new I/O device and other known good devices. The tests have been refined and expanded since the workshops began. The testing is now at a mature level for full and low speed USB devices.

Testing at the Compliance Workshops are only avaialble to members of the USB-IF. The USB-IF maintains a list of products that have passed the testing known as the Integrators List. It is available as a reference list of quality products to members of the USB-IF. Information on the next USB-IF Compliance workshop can be obtained at http://www.usb.org/developers/complian.html

In addition to Compliance Workshops, the USB-IF full and low speed testing is available through several certified USB-IF test houses. A list of the test houses is maintained at the USB-IF website.

In 2000 the USB-IF introduced two new trademarked logos that can be used by devices, hubs, and host controllers that successfully pass the USB-IF Compliance testing and sign a logo license agreement. The logo has two versions: one version for full and low speed and one version for high speed USB devices. The USB-IF intends to heavily market the new logos to buyers in the channel as a distinguishing mark for quality products. The logos are shown in Figure 8-3.

Figure 8-3. New USB Logos will be used in 2000 and beyond

The USB-IF is in the process of developing new testing for the high speed version of the logo. This testing will build on top of the current set of tests for full and low speed devices. To help ensure a positive end user experience, one of the USB-IF requirements for the high speed logo is that high speed capable USB 2.0 devices also are fully functional in full speed environments. Therefore, all of the existing full and low speed testing applies directly to high speed devices. The testing procedure for the current USB-IF full and low speed tests is well documented. A link to the procedure can be found at the following USB-IF website: http://www.usb.org/developers/complian.html

The additional testing that is being added for USB 2.0 focuses on additional electrical tests. The USB 2.0 specification defines a set of new electrical test modes that high speed capable devices must support. These test modes allow extremely thorough and repeatable high quality electrical measurements of the device signaling and receiving capabilities to be measured.

Testing for the high speed version of the USB-IF logo will first be available at the January, 2001 Compliance Workshop. A technical overview of the current full and low speed USB-IF testing as well as a preview of the anticipated set of tests for the new high speed logo are presented in the following sections. The actual tests for both logo programs are subject to change.

USB-IF Testing Details

One of the USB-IF requirements for the high speed logo is that high speed capable devices also function normally in full speed environments. Therefore all of the current USB-IF full and low speed tests still apply. This overview starts with a summary of the current tests. Detailed information and test procedures for these tests can be found at http://www.usb.org under the developers section of the website. In particular, a detailed document going through the setup and test procedure for full and low speed USB-IF testing is available at: http://www.usb.org/developers/data/compliance

At the time of writing the document posted was USBIF-TestProc09.pdf and is included on the CD-ROM; please check the web site for Version 1.0.

Interoperability

There are two types of Interoperability testing that are performed through the USB-IF. The first is general interoperability between devices and systems that are brought to the workshops to be tested. Many of the device vendors who attend the workshops also set up system suites with their devices. Many of the other workshop attendees try to visit every one of these suites during the conference to verify that their devices function with other devices in systems. A device must pass 80% of these pure interoperability tests (not cause problems for other devices) to obtain the USB-IF logo. This testing between a wide variety of different types of devices from many manufacturers allows problems that might otherwise go undetected to be discovered and resolved before products ship. It is one of the most valuable benefits of attending Compliance Workshops. Another

valuable benefit of attending the Compliance Workshops is that expert USB-IF engineers are available to help diagnose and debug issues that arise during testing. Much of the actual electrical and device framework testing that is performed at the Compliance Workshop is performed by volunteers. Many companies have found that sending one or more volunteers to a Compliance Workshop to learn about and perform the testing firsthand is an excellent way to prepare for attempting to certify a product for the first time.

The other part of the interoperability test is a standardized test involving a fixed "gold" tree of USB devices. The devices in the tree change periodically but the basic gold tree topology is shown in Figure 8-4.

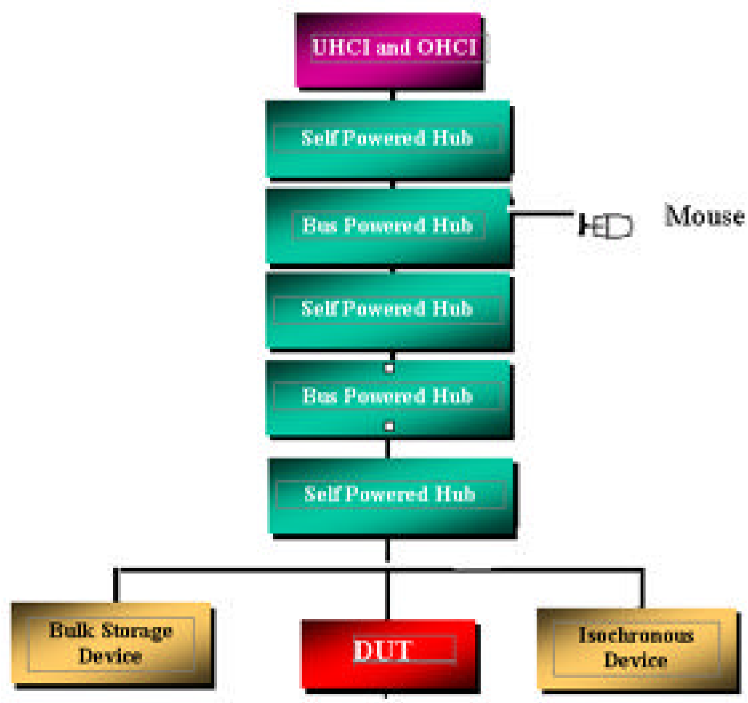

Figure 8-4. A GOLD tree is used for the Device Under Test

The host system runs Windows 98 SE and contains both UCHI and OHCI host controllers. Both host controllers are used as part of the interoperability testing. The basic steps to the interoperability test are as follows:

1. Test that the device under test enumerates and functions properly behind five levels of hubs. The hubs are connected behind the OHCI host controller.
2. Test that the device under test enumerates and functions properly behind the UHCI host controller.
3. Test that the device under test installs all of its software without requiring a reboot when it is first connected to the clean system. No additional software should need to be installed for the device to be fully functional before or after the normal installation process when it is first connected to the system.
4. Test that the device (and other devices) in the gold tree function normally while all the devices in the tree including the device under test are operating simultaneously.
5. Test that the device under test does not prevent the system from executing an S3 system sleep cycle while the device under test is operating. The device under test must allow the system to go to execute a sleep cycle normally while it is connected but not actively functioning. There are a few exceptions to the requirement to allow an S3 sleep cycle to happen normally while the device is active, but they are handled on a case by case basis by the USB-IF.
6. The device under test must continue to work normally after it is disconnected and moved to a different location in the gold tree.
7. The device under test must continue to function normally after cold and warm reboots.

Interoperability Additions For High Speed Testing

The basic interoperability tests for the high speed logo are not finalized, but will likely be very similar to the current full and low speed tests. The gold tree will be redefined to include high speed devices using each of the USB transfer types. High speed capable hubs will be used at each of the first four tiers in the tree, with a full/low speed and a high speed hub at the fifth tier. High Speed devices will be tested below both hubs at the fifth tier of the tree.

USBCheck

USBCheck is a software tool that tests device compliance with the standard device commands and descriptors defined in Chapter 9 of the USB specification. It also tests Chapter 11 compliance for USB hubs and tests compliance with the HID specification for Human Interface Devices. USBCheck along with help describing the test areas it covers is included on the CD-ROM in the TOOLS directory, it can be downloaded from http://www.usb.org/developers/tools.html.

USBCheck functions by loading a diagnostic device driver for the device when it is connected to the system (USBCheck must be running first) instead of the device under test's normal driver. USBCheck is currently supported only under Windows 2000. A screen shot of the initial USBCheck screen after a device is connected is shown in Figure 8-5.

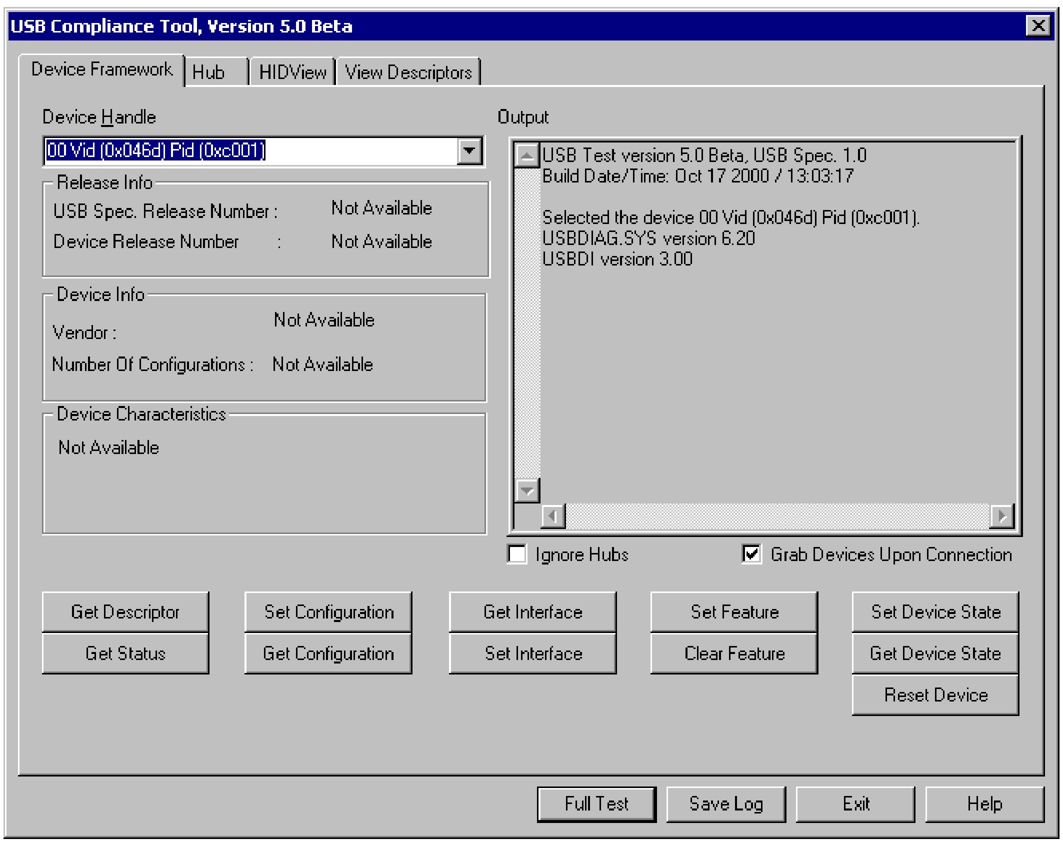

Figure 8-5 USBCHECK is used to test basic operation of an I/O device

Once the option is chosen to run the Device Framework tests (Full Test Button), USBCheck brings up a screen displaying all of the Chapter 9 tests. All of the tests are selected by default as shown in Figure 8-6.

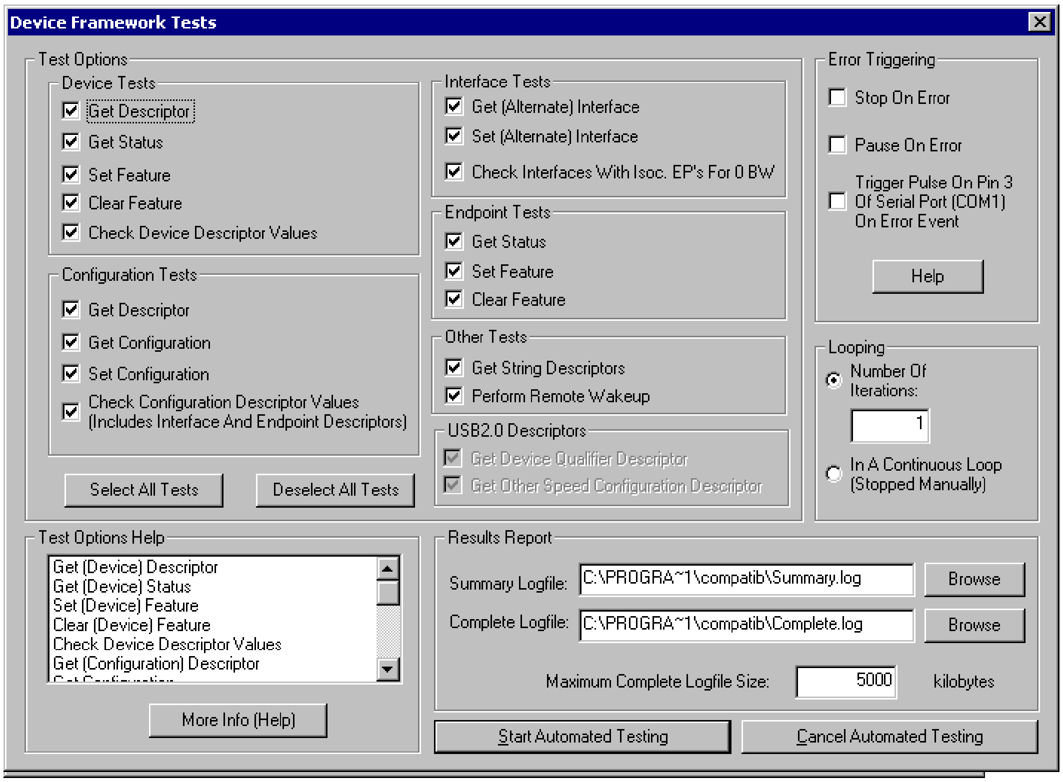

Figure 8-6. Chapter 9 tests the Enumeration Sequence

The Chapter 9 test examines all of the device and configuration descriptors for compliance with the Chapter 9 specification. If the device indicates that it supports remote wakeup functionality it must be able to generate a remote wakeup event. This requirement sometimes is an issue for device vendors at Compliance Workshops. Vendors must bring the necessary equipment to cause their device to indicate a remote wakeup event. USBCheck is easy to download and run. USBCheck is also used by some of the Microsoft test programs for USB devices. Be sure to download and run USBCheck before attempting either USB-IF or Microsoft WHQL certification for a full or low speed device.

USBCheck Additions For High Speed Testing

High speed capable USB devices will be required to run and pass USBCheck testing while operating as both full and high speed devices. The main additional requirement for USB high speed capable devices that USBCheck examines is the proper implementation of the Other Device Qualifier and Configuration Descriptors. The USB-IF requires high speed capable devices to also function normally in full speed environment. The Other Device Qualifer and Configuration Descriptors provide a mechanism for the device to report its capabilities while operating at the "other" speed. Thus while operating at full speed the device should report its high speed configuration descriptors with the Other Device Qualifer and Other Configuration Descriptors and vice versa.

Note: The device should only report its configuration descriptors for its current operating speed through the standard device and configuration descriptors.

This allows, among other things, software to determine when a high speed capable device is connected to a full speed port. If software detects that there are high speed capable ports available in the system it can let the user know that better performance could be achieved by relocating the device. USBCheck also checks new encoding rules for endpoint descriptors. The allowed values for bInterval and wMaxPacketSize are different from full and low speed devices in some cases.

Full and Low Speed Signal Quality

The USB-IF tests device signal drivers to ensure that they meet specification requirements for voltage levels and crossover times. Part of the signal quality testing is also to perform functional testing of receiver sensitivity by ensuring the device is functional in the largest systems (five hubs deep) allowed by the USB specification. Hubs with marginal signal quality are used intentionally to make this functional part of the test as stressful as possible. For the signal qualitity measurements software is used to generate traffic from the device in response to standard commands such as get configuration. The device is connected to the system via a test fixture that allows an oscilloscope to measure the traffic on the data lines.

This data is used to produce what is known as an eye pattern diagram for the transmitted signal quality. An example full speed eye pattern is shown in Figure 8-7.

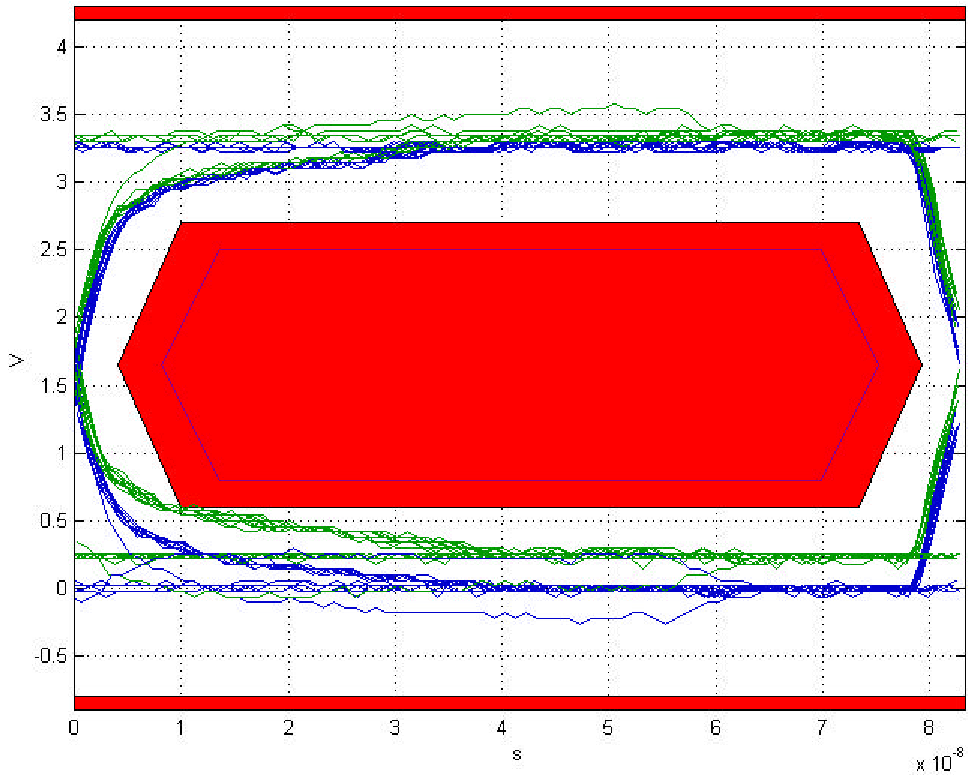

Figure 8-7. Eye pattern for a full-speed I/O device.

If the signaling voltages are two high or too low they will impinge on the central eye pattern resulting in a violation. In addition, if rise or fall times are too slow or vary too widely the eye pattern will be violated. The matlab scripts used to analyze data to produce the eye pattern diagrams are available on the CD-ROM and from the USB-IF website.

The USB-IF test procedure document details the test fixture and setup used for making the measurements. It also indicates which test equipment has been tested and is supported by the MatLab scripts for making the measurements. Figure 8-8 shows some of the test fixtures provided to members of the USB-IF.

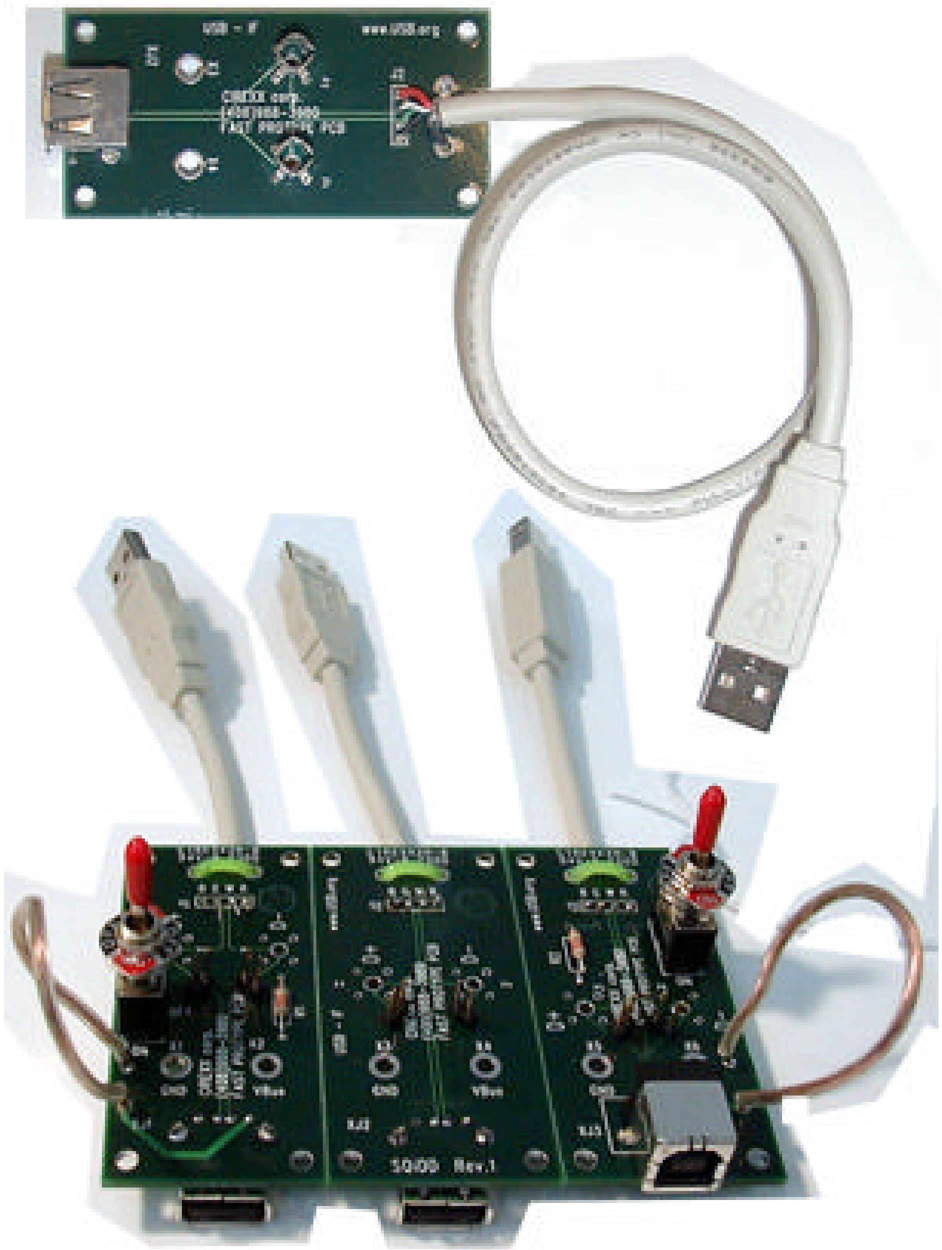

Figure 8-8. Some of the test fixtures provided to USB-IF members

Inrush Current

This test measures the Inrush current that is consumed by the device when it is first connected to the system. The USB-IF testing uses a test fixture and oscilloscope current probe to measure the inrush current. It is measured in a period from 500 mS to 1 second after the device connection. During this period the device is not allowed to consume more than 50.0 mC. Detailed information on the equipment, setup, and procedure used for measuring the inrush current is documented in the USB-IF Full and Low speed test procedure on the CD-ROM and downloadable from http://www.usb.org/developers/data/compliance

Power Consumption

This test measure the device current consumption in several different states. It measures the consumption in the unconfigured state, the configured state, the operating state, and the suspended state. The requirements for these measurements are as follows:

Unconfigured:	Less than 100 mA for all devices.
Configured:	Less than 100 mA low powered, 500 mA high powered
Operating:	Less than 100 mA low powered, 500 mA high powered
Suspended:	Less than 500 uA, 2.5mA for remote wakeup devices.

Note: For high speed capable devices these measurements will be made in both high speed and full speed electrical environments. In addition to these maximum values the device must not consume more current then it reports through its device configuration descriptor(s). This value is read and recorded during the Device Framework testing that USBCheck performs.

High Speed Electrical Testing

The major addition to the USB-IF testing for high speed capable devices is a series of new more thorough and repeatable electrical tests. To enable thorough and repeatable electrical testing the USB 2.0 specification describes a set of test modes that must be supported in the silicon. Once a device has been placed in a test mode the only way to exit the test mode is to power cycle the device. This allows the device to be isolated from the host once it has been placed in a test mode. All information provided on the USB-IF high speed electrical tests and procedures that follows is preliminary and subject to change.

Note: The device can not depend on seeing SOFs to stay in the test mode. Once in the test mode the device must stay in that mode. The only way it can leave it is due to device power being reset.

The USB 2.0 test modes are documented in Section 7.1.20 of the Universal Serial Bus 2.0 Specifcation. The USB 2.0 electrical test modes are described here as follows:

Test_J: In this state the device must drive a continuous high speed J state on the port. This allows the J voltage level to be accurately measured with a voltmeter.

Test_K: In this state the device must drive a continuous high speed K state on the port. This allows the K voltage level to be accurately measured with a voltmeter.

Test_SE0_NAK: In this state the device must enter the high speed idle state. In addition, the device must respond with a NAK to any IN packet sent to the device. This is an unusual state for the device since it must respond to IN packets without necessarily having SOFs present on the bus. The NAK responses allow the receiver sensitivity of the device to be tested. The high speed idle state allows the impedance of the high speed signaling path to be measured.

Test_Packet: In this mode the device must continuously broadcast a long packet defined in section 7.1.20 of the USB 2.0 specification. The packet is designed to cover a wide range of possible device signaling. This packet allows the signal quality of device transmission to be measured accurately in a very deterministic fashion.

High Speed Electrical Test Requirements:

A variety of test equipment is required to perform the high speed electrical tests. The equipment includes:

1. Data Generator
2. TDR (Time Domain Relectometer) (May not be required)
3. Oscilloscope
4. Voltmeter
5. GPIB card.
6. Differential voltage probe.
7. Single ended voltage probes.

The USB-IF intends to publish a list of the specific equipment it has successfully used for the high speed testing on the website in the future. The high speed testing will be offered at Compliance Workshops starting in January, 2001. Several independent test houses are expected to be certified to perform the testing by the end of 2001.

Test Fixture

In order to perform accurate reproducible measurements a test fixture(s) must be used in performing the electrical tests. The test fixture is used to connect the device to a host in order to place the device into a test mode. Once this is done a relay on the text fixture is used to isolate the device from the host. Oscilloscope probe connections and/or SMA connectors for a data generator or TDR are then used to make the appropriate electrical measurements. Each of the new high speed measurements are described briefly in the following sections. The fixture has the following basic requirements:

1. Ability to isolate device under test from the rest of the USB bus.
2. Ability to provide VBUS to the device under test.
3. Appropriate connectors for a data generator and TDR.
4. Four layer impedance controlled PCB
5. Differential pair impedance requirement: 90 Ohms +/- 15%
6. Ideal termination resistors: 90 Ohms, 1% (or two 45 Ohm, 1% resistors)
7. No stubs greater than 0.15 inches
8. Over all fixture trace length <3 inches (from connector to ideal termination)
9. Test probe type: differential active fet probe, <1pf

The USB-IF plans to make fixtures available in 2001 through their website.

High Speed Signal Quality.

High speed signal quality is measured on the device's upstream port. For USB 2.0 the eye patterns and the points at which they are a measured are explicitly defined in the specification. The different test points defined in specification are as shown in Figure 8-9.

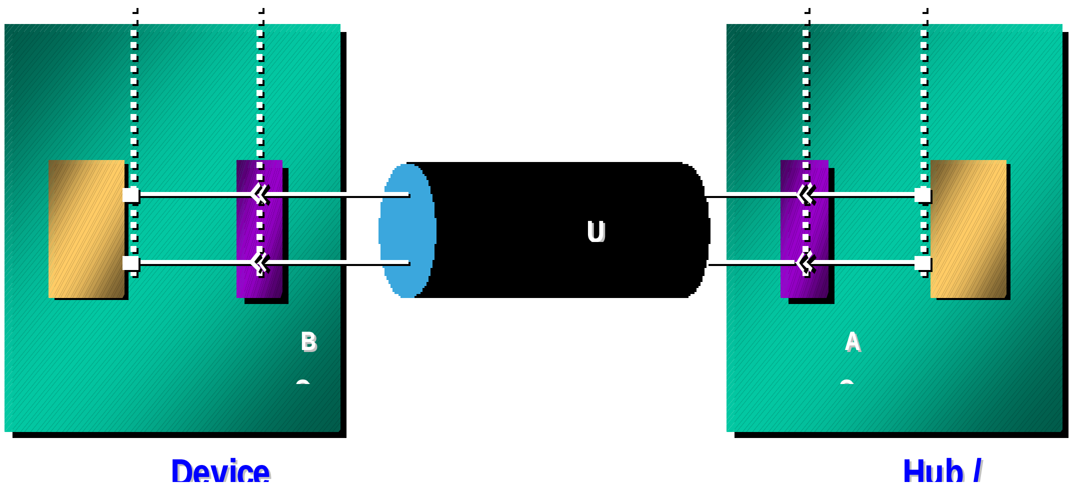

Figure 8-9. High speed test points are defined in the USB Specification

The signal quality eye patterns obtained from the measurements must agree with the transmit eye patterns for Test Point 3 defined in the USB 2.0 Specification. The basic procedure involved in performing the high speed signal quality measurements is as follows:

High Speed Signal Quality On Upstream Port

a. *Place device in Test Mode TEST_PACKET.*

b. *Isolate Device Under Test From System While Maintaining Bus Power.*

c. *Measure Transmitted WaveForm With High Speed Oscilloscope and Differential Probe through Ideally Terminated Test Fixture. Make measurements at TP3.*

   Testing Equipment: Tektronix TDS694c Oscilloscope and P6247/P6248 Differential Probe.

d. *Generate Eye Pattern Diagrams from Data.*

e. *Compare With USB 2.0 Specification Transmit Eye Patterns.*

f. *Check Rise/Fall Times To Make Sure They are not faster then minimum set in USB 2.0 specification.*

g. *Check for non-monotonic signaling transitions.*

A sample of an eye pattern generated from the following procedure is shown in Figure 8-10

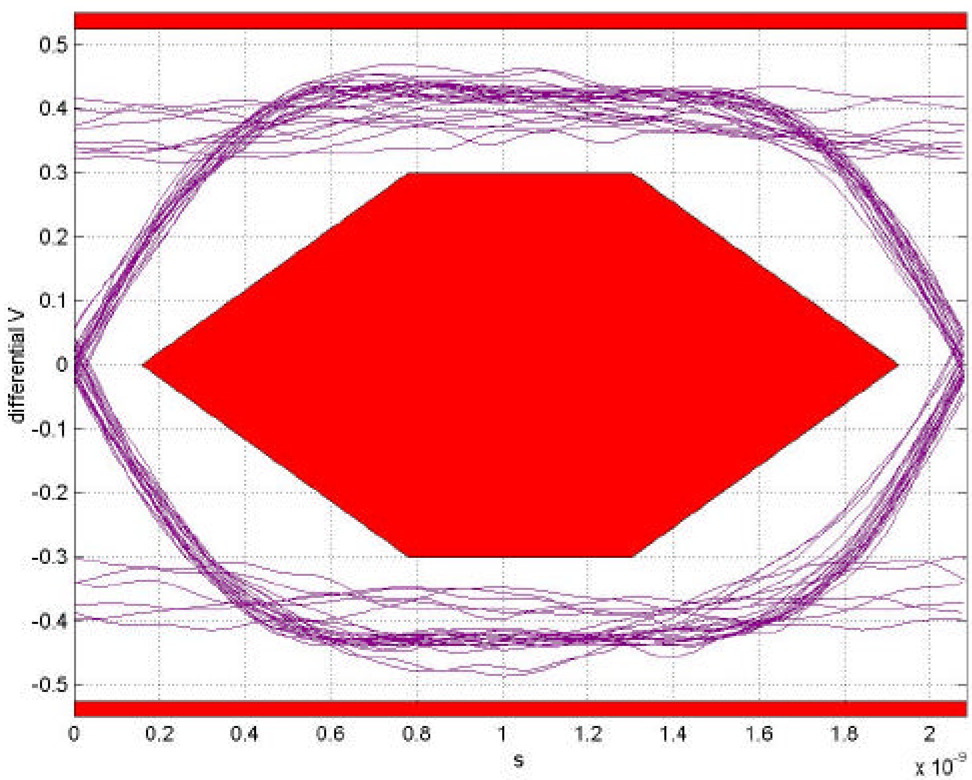

Figure 8-10. Eye pattern for a high-speed I/O device

Any transitions which impinge upon the eye, have a rise time faster than the minimum allowed by the specification, or which do not rise or fall monotonically create failures. There are a number of factors that can contribute to problems with high speed signal quality. Most of these issues are related to problems during board layout. It is possible for there to be issues with the silicon itself. Device silicon providers can provide reference designs to the USB-IF. If the reference designs pass the USB-IF high speed testing the silicon provider can use the logo in marketing their product. If you are concerned with your device receiving USB-IF certification you may want to consider choosing a silicon provider that has used their silicon to obtain certification with a reference design.

Receiver Sensitivity

Receiver sensitivity is measured at both upstream and downstream ports. The basic test procedure for an upstream port and the criteria that will be tested as documented by the USB-IF are as follows:

Upstream Port

a. *Place device under test in test mode SEO_NAK*

b. *Isolate device under test from USB Bus.*

c. *Connect Data Generator to Data Lines*

   Testing Equipment: Tektronix DG2040, TDS 694C, P6247/6248 Differential Probe.

d. *Connect Oscilloscope*

e. *Generate IN Packets of steadily decreasing amplitude.*

f. *Verify that all packets are NAK'd that are above the required voltage threshold.*

g. *Verify that no packets are NAK'd that are below the squelch level.*

h. *Generate sequences of packets that approximate some of the worst case conditions in the required receiver sensitivity eye patterns.*

   1. *Jitter*
   2. *Rise and Fall Times.*
   3. *Voltage Levels*

i. *Verify these packets are still NAK'd*

Receiver sensitivity testing was only done functionally for full and low speed devices. This was done by making sure the devices functioned below a chain of five hubs with borderline signal quality on transmissions. For high speed devices the SE0_NAK test mode allows deterministic testing of the device receiver. The USB 2.0 specification defines a minimum signaling amplitude above which the device must recognize IN packets and respond with a NAK. The USB 2.0 Specification also defines a squelch level below which the receiver must never respond to prevent spurious noise events from being detected as signaling. These values are defined at 150 and 100 mV respectively. In addition to checking these two values, the receiver sensitivity testing attempts to approximate IN packet signaling having characteristics of the worst case eye pattern that receivers must recognize. This eye pattern is defined in the USB 2.0 specification.

J and K Voltage Levels

The Test_J and Test_K test modes are used to take simple voltage measurements for static J and K signaling states. These states allow a voltmeter to be used for precise measurements of the J and K voltages. The procedure for making the J and K voltage measurements are as follows:

a. *Place port in Test Mode TEST_J*
b. *Measure Accurate J Signal Level With Voltmeter*
c. *Place port in Test Mode TEST_K*
d. *Measure Accurate K Signal Level With Voltmeter*

Time Domain Reflectometry (TDR)

TDR testing provides a way to look at the spatial impedance of the high speed signaling path and active terminations of the device under test. The requirements for TDR testing are set forth in section 7.1.6.1 of the USB 2.0 Specification. A TDR works by sending a weak electrical pulse to the device while it is in the high speed idle state. By measuring the reflections of the signal over time the TDR is able to calculate the impedance of the device and its terminations at each distance along the signaling path. Poor impedance matching can result in receiver problems, even though the signal quality on transmissions may meet the specification. The procedure for performing TDR measurements is briefly as follows:

a. Place device under test in test mode SE0_NAK

b. Isolate device under test from system while maintaining bus power.

c. Verify there is no activity on data lines.

d. Connect TDR and perform measurements.

Testing Equipment: Tektronix TDS8000. TDS 694C, Voltmeter.

A sample TDR measurement is shown in Figure 8-11.

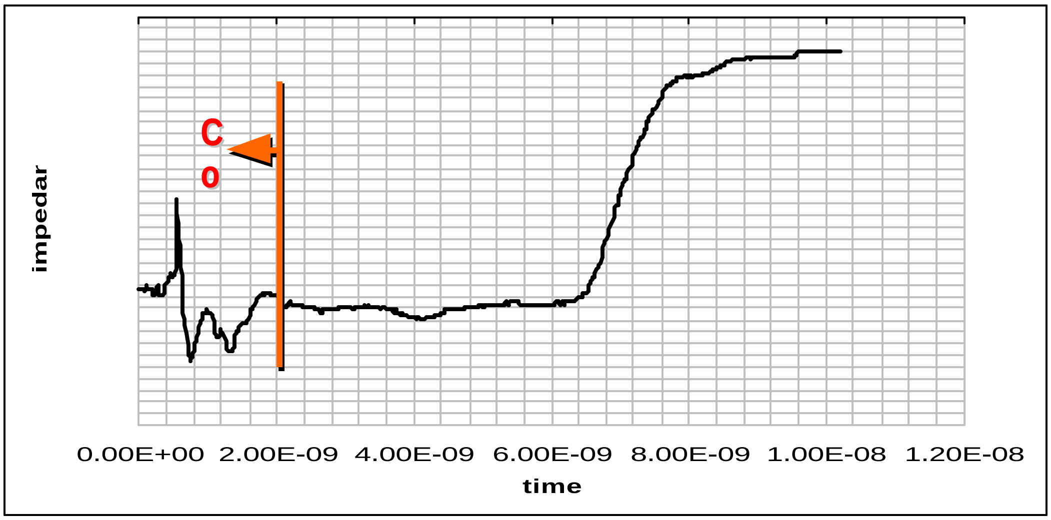

Figure 8-11. A sample TDR Measurement.

When using the TDR, a first run must be used with the cable to the TDR disconnected from the port of the device under test. This allows the time to the device connector to be determined. Another run is then performed with cable connected to the device under test in test mode SE0_NAK. At 4ns after the B connector location the impedance must be between 80 and 100 ohms. From 500 picoseconds before the connector reference time to the device terminations the impedance must be between 70 and 110 ohms. The measurement should be taken with the TDR rise time set to 400 picoseconds. In an up to 1.4 ns contiguous range the impedance is allowed to deviate from the 70 to 110 ohm range as long as no single deviation is for greater than twice the 400 picosecond rise time.

Chirp

When a high speed capable device first connects to a system it must connect as a full speed device. A high speed capable host controller will then initiate a handshaking sequence to determine if the if the device is high speed capable. This sequence is known as the chirp sequence. The basic procedure involved in chirp testing is as follows:

Testing Equipment: TDS 694C and Two Single Ended FET Probes (P6243 or P6245)

a. *Connect Device Under Test To Known Good Host.*

b. *Measure electrical connect event with single ended probes on both Data Lines.*

A sample of a chirp handshaking event is shown in Figure 8-12

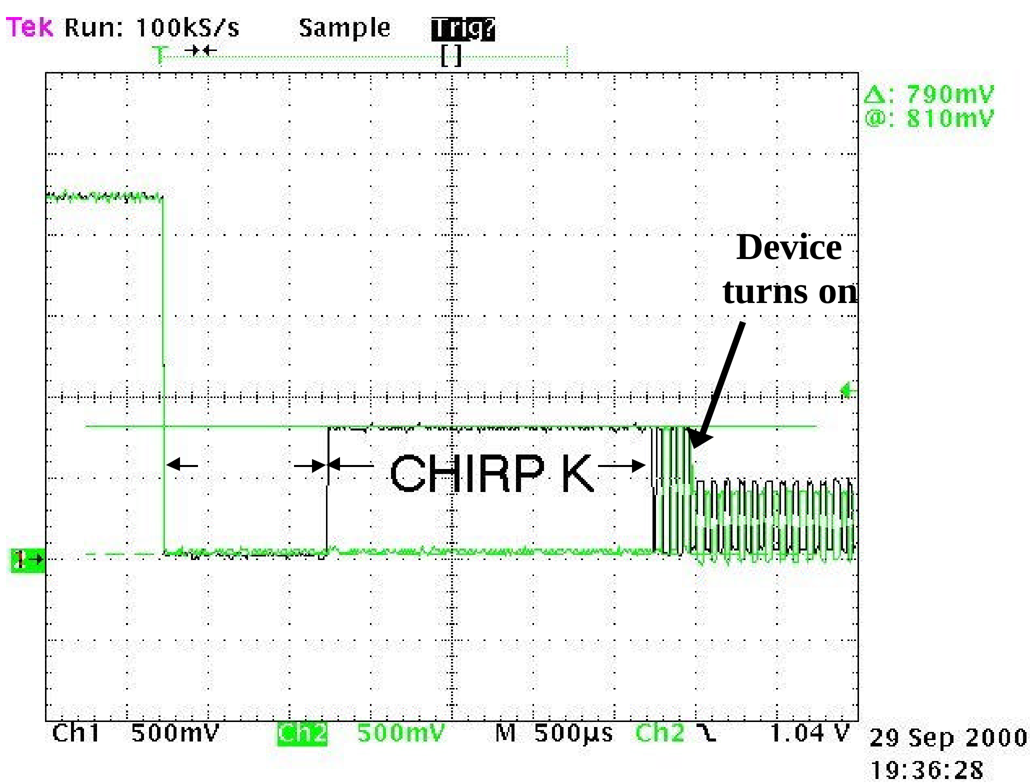

Figure 8-12. I/O device responding to a Chirp

After the device sees the reset it must transmit a chirp handshake no sooner than 2.5 us and no later than 3 us after the reset. The chirp must last between 1 and 7 ms. After the device completes the chirp signaling the host sends a sequence of KJ pairs. The device must be able to recognize the KJKJKJ sequence and apply its high speed terminations. The following parameters are examined during the testing.

1. *Full Speed Idle Voltage*
2. *Duration from Reset To Chirp K*
3. *Chirp K Amplitude*
4. *Chirp K Duration*
5. *Number of KJ Pairs Before HS Terminations Applied*
6. *Delay after KJKJKJ before Device Applies HS Terminations*

Packet Parameters

There are several important packet characteristics for upstream signaling that will also be examined as part of the USB-IF testing. They include the device response time and the number of sync and EOP bits in device transmissions.

MICROSOFT WHQL USB TESTING OVERVIEW

Microsoft Windows Hardware Quality Labs testing of USB devices focuses on the device driver functionality with Microsoft Windows operating systems. If a USB product is shipped with a USB system that has obtained Microsoft WHQL certification the USB product must also be certified. For example, a USB keyboard that ships with a WHQL certified PC must have obtained WHQL certification for the overall system to be certified.

Information on the WHQL certification requirements for USB devices can be obtained from http://www.microsoft.com/hwdev/winlogo/usb/

Since the Microsoft testing focuses on software driver qualitity it complements the USB-IF logo programs. The USB-IF focus is on overall electrical signaling and protocol behavior of the device. A technical overview of the current contents of the Microsoft WHQL test program for USB devices is given in the following sections. This information is subject to change.

MICROSOFT WINDOWS HARDWARE QUALITY LABS TESTING DETAILS

Microsoft testing focuses on ensuring high quality device firmware and Windows device drivers that are interoperable with Microsoft operating systems. Microsoft generally releases a new version of the WHQL test kit slightly before each new release of an operating system. The next release of the WHQL test kit for a consumer version of Windows 2000 is expected sometime in early to mid 2001. Frequently, new USB tests are added to the USB device certification process with these releases. A preliminary look at some of the USB tests expected in this release follows: Note, this list is provided only for reference and is subject to significant change.

Isochronous Interfaces: Devices that consume isochronous bandwidth can be bad citizens on the bus by having bandwidth requirements in their default interfaces. This test checks all of the device configuration descriptors to ensure that interfaces that consume isochronous bandwidth provide alternate settings and do not consume isochronous bandwidth in alternate setting zero (either through having no isochronous endpoints, or isochronous endpoints with zero bandwidth requirements in this setting).

Device Class Compliance: Examines device descriptors to verify that they interfaces comply with current specifications if the device indicates that it belong to a USB device class. For information on the various device class specifications refer to http://www.usb.org/developers/devclass.html.

Port Reporting: Microsoft operating systems contain troubleshooting wizards that attempt to help USB users in situations where devices are not functioning properly. One of the ways that they do this is by instructing the user to try another port when a device is not functioning correctly in the port to which it is connected. If a hub with an embedded function is present the wizard will determine the number of ports available from the bNbrPorts field in the hub descriptor. Often hubs with an embedded function do not expose some of their ports. If these ports are still reported through the bNbrPorts field the MS wizard may provide the user with incorrect information. The hub/embedded device must not report unconnected unavailable ports.

Device Installation and INF Syntax Checker: There is a program that will parse the INF file used for installing the device to check it for proper syntax. USB device installation must not require a reboot. USB device installation also must not require the user to install additional software either before or after connecting the device to the PC. All necessary installation should occur through the INF when the device is first connected to a system.

USB Driver Verifier: This tool is an extension of the general Microsoft driver verifier that specifically looks at a variety of USB specific device driver behavior. It attempts to catch potential memory leaks and USB specific parameters errors by monitoring the behavior of a USB device driver. Some examples of the driver behaviors that USB Driver Verifier monitors are as follows:

1. Checks to make sure client driver submits URBs only after IRP_MN_START_DEVICE has been processed.
2. Checks to make sure the client driver only submits URBs when the device is in the fully powered state.
3. Checks the length of all URBs.
4. Checks to make sure the driver does not attempt to reuse a URB or IRP still associated with a pending request.
5. Checks to make sure the client driver doesn't try to modify information in a pending URB.
6. Checks to make sure that a client driver cancels all pending URBs before processing the IRP_MN_REMOVE_DEVICE IRP.
7. Checks to make sure that valid pipes are associated with all transfer URBs.
8. Checks to make sure that the client driver does not attempt to reset any pipes that have pending transfers.

Serial Number Test: Devices are required to use unique serial numbers.

Driver Stress Tests: Several tools are used to stress USB device drivers (and the system in general). They include programs that repeatedly disable and enable the USB host controller and repeatedly remove and refresh the host controller and devices from the Device Manager. Devices and their drivers most be able to handle these behaviors and continue to function normally. Another test repeatedly suspends and resumes a system to and from the S3 sleep state. A device and its driver must allow and continue to function through this behavior.

Minimum USB Specification Version: The PC 2001 System Design Guide requires that all USB devices be implemented to version 1.1 of the USB specification or later. This test examines the device descriptors for the self reported USB specification version. It also examines differences between the 1.0 and 1.1 specification to make sure the device firmware is updated for the USB 1.1 specification version or later.

Device and Hub Power Consumption: This test examines all device configuration descriptors to check that reported device power consumption is 100 mA or less for self-powered devices and 500 mA or less for bus powered devices. Hubs are also tested to make sure they report themselves as self-powered. The PC 2001 System Design Guide requires that all hubs except those integrated into USB keyboards be self-powered. This requirement was put in place to help eliminate end user confusion over all USB ports not being equal.

USBCheck: Microsoft WHQL test kits also sometimes include the USB-IF tool USBCheck described in the preceding technical overview of the USB-IF testing. This tool examines USB hub and device descriptors and behavior for Compliance with Chapters 9 and 11 of the USB Specification.

COMMON DEVICE PROBLEMS UNCOVERED DURING TESTING

This section takes a brief look at some of the most common problem encountered by devices that undergo USB-IF and WHQL certification. It is by no means a complete list, but it shows the kinds of problems that are uncovered by the testing. This list will be extended on this book's companion web site.

Full Speed Signal Quality A variety of devices have problems with full speed signal quality due to the use of common mode chokes. These chokes generally degrade full speed signaling, particularly the full speed SE0, to the point where many devices using the chokes do not pass full speed signal quality. If you are using common mode chokes in your design make sure that you explore the affect they are having on full speed signal quality.

bcdUSB Field There has been some confusion over how devices designed to the USB 2.0 specification should set the bcdUSB field in its device descriptors. Regardless of the device speed (HS/FS, FS, LS) if the device is designed to the USB 2.0 specification it must use 2.0 in the bcdUSB field. The USB 2.0 specification has some FS and LS requirements that differ from the 1.1 specification.

Port Reporting A device that is integrated with a USB hub often has issues with how the hub reports its ports. Microsoft operating systems contain troubleshooting wizards that rely on accurately knowing the number of free available ports in the system. In some cases the compound device does not make all of the ports on the hub available to the user. In these cases the device will fail

WHQL testing if it still reports the unavailable unused ports in the bNbrPorts field of the hub. The Microsoft troubleshooting wizard can provide inaccurate information to the user about alternate available ports to try in this case.

USB 2.0 Test Modes Some high speed capable devices fail to support some or all of the USB 2.0 test modes. High speed capable devices must correctly implement all of the test modes to receive USB-IF certification.

CHAPTER SUMMARY

Adding power management features to a USB I/O device is straightforward and is the first step to becoming compliant with the USB specification. There are several test suites available which will enable you to gain confidence in your I/O device product. There are several benefits to becoming a USB Implementers Forum member including rigorous product testing at a Plugfest, inclusion in the Integrators List and marketing exposure at industry events. The USB Implementers Forum is in place to promote USB adoption and products and, as their membership increases, they are a larger influence on the marketplace.